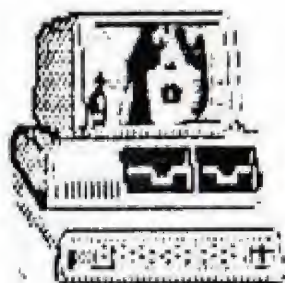


Adventure Coder

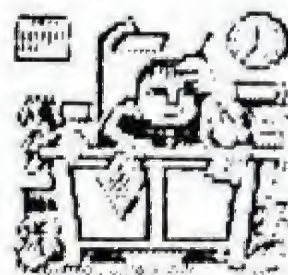
JULY 1989 • ISSUE 1 • £1







CONTENTS



- 4 Editorial
- 6 News
- 7 Utilities and add-ons
- 8 GAC+ Review
- 11 Computer poem
- 12 Paw Prints
- 15 Food Chart
- 16 Machine Coding your adventures Part 1
- 19 Crossword
- 20 Whatever happened to...?
- 21 Next month...
- 22 GAC to the drawing board
- 26 The Burning Man
- 28 Useful addresses
- 30 Just A Rollin' Ball
- 31 Adventure Coder needs you!
- 32 Subscriptions page

Front cover artwork "Philosophy" (C) 1982/1989 C Hestel.

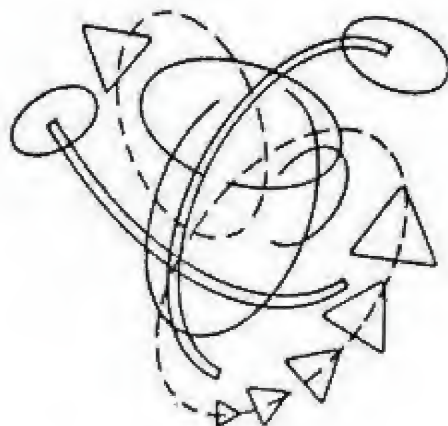
Editorial dreams

When I got my first computer, easily the most interesting use I found for it was to write my own game. I had very little knowledge of BASIC and none at all of other computer languages, so it was with great delight that I read of a new program being released to enable anyone to write their own adventure games, with no programming knowledge required whatsoever. This was a major breakthrough and changed the face of home computing almost overnight. The program was "The Quill" from Gilsott, of course. Before it's release, adventure games had been the sport of players, not writers, unless you knew your machine inside out. After "The Quill" adventure game writing became possible for everyone who owned a Spectrum 48K, no matter what their knowledge of programming. So? That meant that you, or me, were able to try our hand at putting together our very own adventures. It no longer mattered what the professionals were up to, if we didn't like their adventures we now had the chance to write our own. The elitism of programming had gone. Joe Bloggs could now write an adventure! This also meant that the tricks of the trade were ours to see - so that's how you set up a hidden object, so that's how you increase the score when you find the object... the fog had been cleared. If "The Quill" had stopped there that might have been enough, but the ball had been set rolling... soon home-based companies sprang up, offering a range of Quill-written games. Gilsott became inundated with daily stacks of Quilled adventures that the public had written themselves. Whilst a majority of these might not have been fit to publish, there remained a cream of Quill games that were, quite literally, up to the standards of professionally written adventures released by other companies. What's more there was occasionally a Quill game so good it was actually better than a lesser release from an established software house. It became clear that there was no magic about writing adventure games - they weren't written by a select band of people trained over many years, they could be successfully written by everyone. What the Quill did essentially was to remove the need to waste time learning how to program, then writing a program to write an adventure - that way you needed years before you could even start your game - with The Quill, it was all done for you. So you were left to concentrate on the important thing, actually creating your game! Setting up an atmosphere, putting in a set of interesting objects to discover, crafting your location descriptions and messages. In other words, writing the adventure itself, not complex and demanding code to enable you to write the game later, if at all! And to create your text, all that was essentially required was the very words themselves! So anyone who could read and write, and use a computer keyboard was now able to create an adventure! Of course it takes a skilled writer to produce a masterpiece, but why should that stop the general public having a go? It needn't, ever since Gilsott released "The Quill".

Where does that leave adventure writers today? Since "The Quill" many other adventure writing utilities have surfaced, all offering various ways of creating your game. "The Quill" was text-only until Gilsott added "The Illustrator", and now many utilities allow text and graphics to be created together, plus a vast array of extensive and exciting features as adventures get more and more clever. The great thing is that what ever the big companies get up to, with their fancy adventure systems, the public is never far behind thanks to all the companies that have released better and better utilities. Even mouse-driven menus and digitised graphics are now able to be part of your adventure games! Before "The Quill" that would have seemed impossible. Whole areas of fascinating programming are being laid wide open to everyone, not kept secret by a few learned programmers.

Adventure writers have been particularly lucky in this respect, whereas arcade games have only had a handful of game-writing programs made available, and those often limited.

I believe that everyone has the ability to produce an adventure game of some sort, but if you want to write anything worthwhile, then you're reading the right magazine! I decided to set up Adventure Coder not for myself, but to help you, the reader. Whatever computer and method you're using to write your adventure game, then this is the publication for you. Only here will you find all the help you'll need to create an adventure of the highest standards. You may have just started computing and want some help in the basic elements of designing an adventure - how best to make a map, use objects and define your puzzles. No problem. Or you may want more advanced help with a complex routine that you can't seem to fathom out and your game won't be the same without it. Again, no problem! Over the coming months I hope to be able to bring you articles on all aspects of adventure writing, even getting your game published. I'm here to help you, but I can only do so if I know what help you need. Feel free to write in about anything you like concerning adventure writing - I've a team of experienced writers ready to tackle your problems no matter how trivial they may seem to you - everyone has to start somewhere, and there's nothing worse than getting stuck with a line of code that someone else might have already sorted out for you. Maybe instead you're that person, who's written what you consider to be a nifty routine you feel might be useful to other coders. Why not send it in and get your routine printed? You'll be helping others and seeing your name in print at the same time. That can only be good, surely? I'm not looking for just routines of course, anything you think may be useful, such as a bug you might have found in your utility, or a trick to make it do things it isn't meant to. There maybe someone who would love to use what you've found, if only they knew about it. And what if you've an idea you wish to discuss? Why not send in a letter? Perhaps there's something about adventures that really annoys you, are they getting too close to Role Playing Games? Would you prefer all adventures to become text-only again, or do you wish more graphics could go in a game? Why not air your views? Adventure Coder is here for you to do just that. I hope you find it useful. If not, why not say so? I've tried to make it as interesting as possible, but I'm looking for a lot more contributions from you, the reader. See if you can think of a way of improving the magazine - I'm sure there's lots of ways I haven't thought of! I'm only the Editor, and this is your magazine. I look forward to any comments on this first issue - enjoy it!

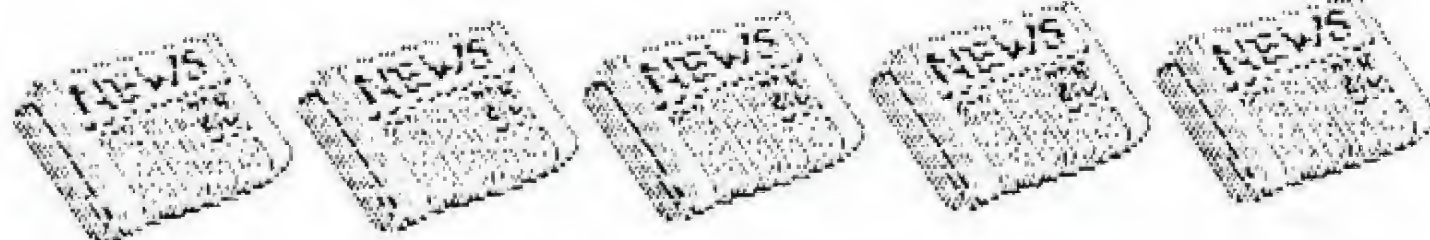
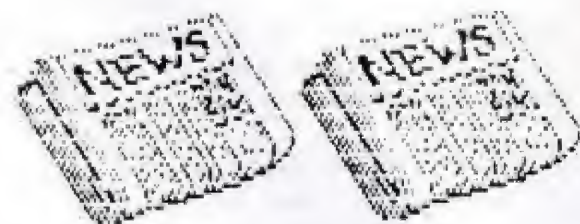


MORE PAW SEÑOR!

Gilsoft are planning to release two further versions of their bestselling adventure writing utility known as PAW (Professional Adventure Writer). The first is a Spanish language version of PAW, apparently because there's a lot of Spectrums in Spain! Even so, there are far more Commodore 64s in the world, which doesn't explain why the 64 version of PAW is still on ice.

Secondly, a PC computer version is being written, which could prove immensely popular, as many machines now support PC software around the world.

Lastly, the ST version of PAW is due out next month... no, sorry, that's a lie. Looks like this one has bitten the ice too.



LICENCE TO QUILL

Other Gilsoft news involves the budget release of The Quill and The Illustrator for three computers, the Commodore 64, Spectrum 48K and Amstrad CPC range. This means at last that The Illustrator has surfaced for the 64. You can get The Quill (for nostalgic reasons?) or The Illustrator alone for £3.99 each, or both for just £5.99. State which computer you own and add 50p p&p if you wish to taste some vintage computing history, from Gilsoft International Ltd, 2 Park Crescent, Barry, South Glamorgan, CF6 8ND.

Utilities and add-ons

This is a list of all known programs but I'm sure there are a few more. If you know of any other utilities or add-ons, especially for computers such as the MSX and Atari 8-bits, whatever, please write in and help make this list a definitive guide.

AMI = Amiga 500, (maybe 1000, 2000 too)
 CPC = Amstrad CPC (464, 664, 6128)
 ARC = Archimedes (305, 310, 310M, maybe A3000 too)
 C64 = Commodore 64 (maybe 64X, 128, 128D too)
 DRG = Dragon 32
 ELE = Electron
 MTR = Master
 S48 = Spectrum 48K (maybe 16K, 48K+, 128K, +2, +3 too)
 ST = Atari 520STFM (maybe 520STM, 1040STF, MEGA 1, MEGA 2, MEGA 4, STE too)

| <u>NAME</u> | <u>COMPAN</u> (<u>COMPUTERS</u>) <u>COMMENT</u> |
|----------------------------|---|
| A-CODE | Level 9 (many) in-house utility only |
| ADVENTURE BUILDER SYSTEM | α Ω (S48) |
| ADVENTURE CONSTRUCTION SET | Electronic Arts (C64) |
| ADVENTURE KERNEL SYSTEM | Melbourne House (AMS) book listing, tape |
| ADVENTURE WRITER | Codewriter (C64) USA Quill |
| ADVENTURESCAPE | A&B (BBC) |
| ADL | Public Domain (AMI) |
| ADVSYS | Public Domain (ST) |
| ALPS | Alpine Software (BBC MTR ARC) |
| AMIGAC | Incentive (AMI) Coming soon? |
| AMIGAVENTURE | Public Domain (AMI) |
| THE BIRD | Ramjam Corporation (many) in-house, to loan |
| CHARACTER SETS | Simicro (S48) GAC |
| CHARACTERS | Gilsoft (S48) Quill |
| DRAGON WRITER | Coven (DRG) |
| DUNGEON BUILDER | Dream (C64) |
| THE EXPANDER | Gilsoft (S48) with PRESS |
| FONT CREATOR | Simicro (S48) GAC |
| THE FIX | Kelsort (S48) |
| THE FIX+ | Kelsort (S48) |
| GAC | Incentive (S48 AMS C64) |
| GAC+ | Incentive (C64) disk-only |
| GAC DATABASE PRINTER | Big Sky (C64) |
| THE GACFAC | Essential Myth (S48) GAC |
| GENESIS | CRL/Camel Micros (S48) good band! |
| THE ILLUSTRATOR | Gilsoft (S48 AMS C64) Quill |
| MEGA | Gilsoft/Kelsort (S48) PAW, part of PTM |
| MINIFIX | Kelsort (S48) |
| PATCH | Gilsoft (S48) |
| PAW | Gilsoft (S48 AMS PC) no C64/ST! |
| PAW-PHOSIS | Gilsoft/Kelsort (S48) PAW, part of PTM |
| PAW-TEL | Gilsoft/Kelsort (S48) PAW, part of PTM |
| PIC-FIX | Kelsort (S48) |
| PRESS | Gilsoft (S48) |
| PTM | Gilsoft/Kelsort (S48) 3 PAW overlaps |
| QUAID | Kelsort (S48) Quill |
| THE QUILL | Gilsoft (S48 AMS C64) |
| RECLAIMER | Kelsort (S48) GAC |
| SAGA | Scott Adams (C64) not for sale! |
| THE SCRIBE | Your Spectrum (S48) listing |
| SIAC | Incentive (ST) |
| TAC | Incentive (BBC ELE) GAC without graphics |
| TAILSFIN | Microdeal (ST AMI) |

GAC+ Review

Chris Hester reviews "GAC+" from Incentive on the Commodore 64.

It all started with the release of the original "Graphic Adventure Creator". This was released by Incentive in 1985 on the Amstrad first, and the Commodore 64 and Spectrum in 1986. Written by Sean Ellis, here was a versatile adventure writing program that stunned reviewers and programmers alike. It was a combined text and graphics program, unlike Gilsoft's separate modules "The Quill" and "The Illustrator", so all you needed to create a decent graphic adventure was load "GAC"! There followed a BBC text-only version of the program too, but I bought the Commodore 64 "GAC" as soon as it was released. Converted by The Kid in the spring of 1986, it proved immensely useful to a CBM64 market that had been deprived of a version of "The Illustrator", and hence a chance to design your own graphic adventures. Although none of the unique characteristics of the 64 were exploited such as sprites and quality sound, "GAC" was well worthy of the Gold Medal awarded to it by "Zzap! 64" magazine. Or was it? Thorough testing by myself revealed a list of bugs and nasty things that could go wrong with the program. Most of these were avoidable thankfully, but they spoiled an otherwise quality product. With the advent of the 16-bit computers, an Atari ST version of "GAC" was released, known as "STAC". This was a far superior version allowing graphics from art programs to be used and different text fonts, hence breaking the "samey" look that was an attribute of the previous versions of "GAC". That brings us up to date with the release of "GAC+" in 1989 for the Commodore 64 only and on disk only as well. But before I can comment on "GAC+" as it is now released, there's a three-year tale to tell of it's inception! Read on in bewilderment...

"GAC Plus", as it was to be called, was conceived soon after "GAC" as a superior version with a host of new features. In early 1986 Amstrad owners were the first to be told of a possible deluxe version of "GAC", to feature a music editor, graphics that could be scaled and merged, and loads of shading patterns, much like "The Illustrator". The new program would even be mouse compatible! Available on disk, you could access other disks for data, just like Intocom, and hence create an infinitely large adventure! "GAC Plus" was to be released in... get this... June or July 1986. That's three years ago!! Later on that same year, Incentive announced further news of this stunning product, by now eagerly awaited. The release date had been changed to September 1986 and "GAC Plus" would then offer the Amstrad user a "perfect fill routine", and text on the graphics screen, plus the screen editor had been improved and now included a disk catalog function, you could have redesigned characters for your text, and room descriptions along with messages could be 1024 characters long! (As opposed to 256.) There would also be a new improved text compression routine. Existing "GAC" users could upgrade their copies for... wait for it... free! Otherwise "GAC Plus" would cost £49.95. All in all, it looked good for Amstrad users in 1986!!

So what happened? That, only Incentive themselves know. It would appear that they came across the idea for their "Freescape" 3D graphics routine and that took over. I would hedge a bet that the Amstrad "GAC Plus" was shelved and other straight versions of "GAC" released instead for other computers. They may have used some of their experience in attempting "GAC Plus" on the Amstrad to write the Atari "STAC" version, but there remained no "GAC Plus" for the 8-bit computers until this year, with the release of what we now know is called "GAC+" on the Commodore 64. Sorry, Amstrad users, you've lost out there! Now let's explore together all the new and exciting features in "GAC+"...

Oh dear. Looks just the same as "GAC". Where's the music editor for a start? The 64 could really have made use of that, doing away with the reeble "beep" created by pressing the "CTRL" key and "G". You could have had a piece of Bach playing at the start of the game, and a piece of Tchaikovsky during play... still, never mind, let's try out the thousands of new graphics shades I've been waiting three years to try out - hope they're as good as those you get with Gilsott's "The Illustrator" - there, you could use certain shades to create whole new colours and rapidly improve your graphics! What's this though? No extra shades? No amazing patterns? No stunning new features to scale graphics?? Nope. Not a bit of it. To tell the truth there's been absolutely no improvements made to the graphics section at all! Well then. That just leaves all the other fascinating features we can use... er... I hate to tell you this, but they're not there. It appears "GAC+" is not the deluxe "GAC" I was looking forward to! Infact it isn't a "GAC Mark II" at all! It's just "GAC" with the ability to link separate parts to make bigger adventures. That's it. But isn't that alone great news? Yes, but we're being conned. A look at the directory on the disk of "GAC+" reveals all we need to know about what "GAC+" really is. The disk bears a directory header of "GAC USA 2". Get that? Not "GAC+" but "GAC USA". Soon after "GAC" first came out, there was talk of an American version of it being programmed to cater for the American disk market. Selling "GAC" in the States was no good if it couldn't make some use of disks. So in the spring of 1987, The Kid returned to the "GAC" code he'd written a year before, and put in the ability to link a great many parts of your adventure on disk. The Americans were bound to love it. To all intents and purposes, the American version of "GAC" is "GAC+". Otherwise why has nothing been done to the code since 1987?? And why have we had to wait until 1989, two years later, to get this "fake" new version?? And why have Incentive never done the decent thing and released a proper upgrade of "GAC" that they set about writing back in 1986??

So there you have it. Incentive have obviously thought what to do next: write the awaited "GAC Plus" after all, or get rid of the American "GAC" and push their sales a bit more, giving the program to an independent distributor (Mandy Rodrigues) and leaving them free to concentrate on what I can only assume is "Freescape III". You see, arcade games take preference over adventure utilities. If anyone else knows any more about this matter, please write in and tell me!

So what exactly is the benefit of owning "GAC+" then? The only benefit is that you can now create your adventure as large as possible! There's no longer any restriction on memory - you can now create up to 252 parts to your game! (Not millions of parts as you might have hoped, but 250 should be enough!) This has the added attraction that each part can feature masses of text and really detailed graphics. If you run out of memory, simply move on to another part! If you wish, each part can be only one location, but with as much text and graphics as will fit in the computer! You could have a different graphic for each object, displayed upon examining it, and really large messages too - even realistic dialogue that before memory wouldn't have allowed. How many ideas have you had before that to implement with "GAC" would have filled the memory and not left any room for a game? Well now you can try them out! How about designing a space game where each part takes you to a unique planet, featuring lots of graphics and locations? At last there's no limit to the memory your game can use!! That, if nothing else, makes "GAC+" worthwhile.

That's the good news. The bad? If you're thinking of expanding a full-size "GAC" game using "GAC+", then you must make sure it doesn't use up too much memory! With "GAC", you had up to 23124 bytes free to play with. But with "GAC+", you only get 22000 bytes free. Obviously

that isn't a problem to a brand new game as you go on to a second part if the memory is full, but I crashed the program by trying to load in a full-size game into "GAC+" - the game was 1124 bytes too large!! There's no warning about this in the manual so I'm telling you now, be careful about your old "GAC" games - make sure they are less than 22000 bytes long before transferring them into "GAC+" - you might have to take out some locations and put them in a separate part, it's up to you. Or you might want to start fresh on a "GAC+" only game. If so, then you'll be making use of the LINK command, which is what "GAC+" is all about. This can link a lot of different parts together in a manner that leaves the player to explore your game as if it were much, much larger than could fit into the computer at one go. If you split your game's map into definite groups or locations, such as those making up a forest, a village and so on, then all you need to do is create each part separately inside "GAC+", then put in the correct LINK commands between the parts! So the player could then walk from the forest to the village, and the disk would load in each part no matter where the player had been before or was set to go. The only requirement here is that objects match between the parts - "a wand" in the forest (part 1) must also be "a wand" when the player crosses over to the village (part 2). So you'll have to enter many objects again in each part, but that's no problem. I'm sure some objects can be used in one part only, such as a grand piano, say, that wouldn't be able to be moved from one part of your game to the next, so in the next part, that object could be an oak tree, or whatever!

Other things to note are that there's an error in the manual concerning the colours used in the graphics section. For some reason, Incentive have changed the green and blue highlighting colours to black and orange. The manual still refers to green and blue though. No problem to "GAC" users upgrading, but it could cause confusion to people getting "GAC+" and being unfamiliar to "GAC". Black and orange highlighting? Yes, that's right. Guess what. Orange and dark blue (the background colour) is one of the most revolting, messy and unreadable combinations known to Commodore 64 users! Why the orange colour?!? Grrr. Black on dark blue is readable but dark. Tut tut. Why do programmers still use bad colour schemes when there are so many clear and readable ones to choose from? I can only assume that The Kid (who programmed "GAC+") has a very clear monitor. Myself I use a clear TV (as good as the Commodore monitors I reckon.) It may seem a trivial point to you, but it gets to me. Imagine if I were using a really poor colour TV - what then? What hope of reading the text in orange on blue then? Programmers must cater for all users. I'd like to see a lot more programs offering not only the chance to alter the screen colours to your liking, but if that's not realistic (as in a multicoloured arcade game) then there should be the option for different colour schemes, as seen in "Spindizzy". And there should also be a black-and-white option for people without colour TVs. If not... why not? It's stupid to assume all computer owners use a flash monitor.

After that grudge, I see there's still no use for the "LAST" bar on the graphics screen. On the Amstrad version, this shows your last command - useful when editing graphics. But they've never coded it on the 64 for some reason. You just get this bar next to the word "LAST". The bar never changes - it's pointless! However, in "GAC+", the colours of the bar have changed in accordance with the shift from green and blue highlights to black and brown. Still meaningless.

Well you're probably thinking I didn't get on well with "GAC+". On the contrary, I quite like it. But it's dreadfully late. The 8-bit adventure market is shrinking yearly as the 16-bit computers take over, so by the time you've written your multipart "GAC+" game, who's going to still own an 8-bit computer on which to play it? You could

have written three years' worth of games by now, had incentive kept to their release dates! Never mind, "GAC+" is here, and you'd best make the most of it! Get out your pencils and draw the biggest map you can think of. There'll be no trouble fitting it in to your game now!

"GAC+" is only available from Mandy Rodrigues, 24 Maes Y Cwm, Llandudno, Gwynedd, LL30 1JE.

Commodore 64 disk-only price: £29.95 - complete with manual.
Upgrade price: £10.00 (with your tape or disk of "GAC")



Computer Dreams

by Alison Pinkett

At the computer, tap-tapping all day,
I dream of places far, far away,
Swaying palm trees on a sun-drenched beach,
It seems so terribly out of reach...
Maybe this computer has a hidden code
That could just take me into another mode.
Just press T for transport and watch the screen.
As figures become places I've never been.
Then through the screen I would simply float,
No ticket needed for a plane or boat.
Suddenly I hear someone call to me,
"Come on, dreamer, it's time for tea."
But as I press "off" and watch the screen clear,
A sound comes from within - it is the sea I hear?

Reproduced by kind permission of Mrs Alison Pinkett/Woman's Weekly.

PAW Prints

by George March

In this first issue, I'd like to first answer the three most asked questions concerning any future adventure writer, which are..

(A) I know I want to write something, but I don't know what, I've no imagination! (B) Either I've got a few ideas, but that's it! Or I've got so many ideas that I don't know when to stop (this happens very rarely mind) and (C) I want to be able to do such-and-such, but I don't know how to do it?

To which I answer.. (A) Rubbish! Everyone's got an imagination, just some peoples are a bit more abundant. So what can you do to get any ideas? Just like me, you can try asking other people what they would do in a certain situation, given the objects you have in your game to do as they wish with! Your local library will help, or T.V, etc, and are a mine of factual information. The T.V and films (my favourites 'James Bond', 'Our man' and 'In like Flint', 'Mcgyver', etc) are brilliant I find for giving me a prod in the right direction, and newspapers and magazines are full of stories about airplane hi-jacks, kidnappings, etc, for story ideas, and even though I can't condone the nicking of other peoples ideas for stories (though the borrowing of routines is usually okay), a spoof adventure (eg, the 'Boggit') is a workable (if not original) idea. But the way I often go about it, is just to first sketch my few ideas down on paper, and to write a short story with them (the shorter the better!), taking any messages, location descriptions, etc, straight from there! (B) A few ideas (or a short story) are all you'll ever need to start any adventure, not too wordy to start with mind, just to get you into the way of things, and as far as having too many ideas, not everyone will be a good one, so don't try putting too many into a game to start with. Why not keep some routines for a second game (or a larger 128 version), after all a game never needs to be very big to be brilliant, 'Behind closed doors' by Zenobi for example, which only has one location! And (C) if you're ever stuck with something, ask someone else! So why not try writing to me, see the address at the end, or care of the 'Coder'. And even if I can't figure out a problem for you, another reader out there maybe able to! But there's never any need to be stuck with an idea, or have to do without! So why not write for help, or with any ideas, routines, etc, which you think could help someone else, so have a heart and help your fellow programmers and myself write a great column!

Right onto the column properly now, there are quite a few things I wanted to be able to do with the P.A.W for my own creations when I first started out, but never could quite handle, as there was nobody to help until now! Which included the printing of available exits on screen, for example in process 4 the routine could be something like this..

```
-- COPYOF 38 110 LET 33 A MOVE 110 SYSMESS A
```

The first action makes a copy of the players present location number into a blank flag (which I've used flag 110 as), the second gives the verb a direction number (I've used the number A as an example), ie, verb number 2 is actually the word 'south'. This next action moves the blank flag with the same room number as the player in the direction given by LET 33 ?, and then, if the flag is able to move in the direction given, the name of the direction is printed. Please note that a copy of the routine above exist for each available direction, just altering the direction number in LET 33 ? to match, and all directions are measured using those available in the connections table. For example..

```
LET 33 14 MOVE 110, etc.
```

Would give a sysmess like 'An open lift door', only if the exits from

the players present room had an exit leading into an elevator, and ENTER was given as verb 14 being said exit, but for example what if there's a locked door which can be opened, then the routine would be..

```
_ _ AT R ZERO 60 SYSMESS B
```

So is the player in room R (eg, where the lift doors are?) and have the doors been opened yet? If not then print sys B.

```
_ _ AT R NOTZERO 60 SYSMESS C
```

Just as before, just giving a different answer if the door has been opened. Sysmess B = 'Some closed lift doors. ' and C = 'Open lift doors. '. You must remember the space after each full-stop, for these exits to be printed as a valid sentence. Now the routine in response which prints these exits on screen might be..

```
EXITS _ SYSMESS D PROCESS 4 NEWLINE DONE
```

Sysmess D = 'Obvious exit(s) are ' again remember the space! This action then prints an sys without a newline, visits the process (as above) which prints the name of the exit(s) available, and then newlines it to look like a proper message. Now this is okay, if the writer only wishes the directions to be printed on request, but what if we want them to be on screen all the time? Well these routines placed in process 1 will do the trick..

```
* _ 0 NEWLINE ZERO 0 ABSENT 0 LISTOBJ NEWLINE SYSMESS D PROCESS 4 NEWLINE PROTECT
```

```
* _ 1 PRESENT 0 LISTOBJ NEWLINE SYSMESS D PROCESS 4 NEWLINE PROTECT
```

The first four lines in * _ 0, and the first two in * _ 1 are just as P.A.W contains normally, with just another line space inserted between any objects printed, and/ or directions printed on screen, sysmess D is just the same as in the EXITS routine above. I've used another newline to print a line space, after the routine visits the process which generates the exit names, just to make it look a bit cleaner, and the protect action forces any inputs to scroll up underneath the location text (or whatever else is on screen). Here's a simple idea for response to allow the player to change the contents of the open/closed door flag.

```
OPEN DOOR AT R ZERO 60 SET 60 OK
```

So if the player opens the door at a particular room (room R), and the door is not opened yet, the routine marks the door as being open and prints the sysmessage 'Okay'. Now the messages used in process 4 must be either MES's on their own, or SYSMESSes, but neither must have a newline, as the routine will look real funny otherwise! And won't be printed as a continues message as they should (eg, 'Obvious exit(s) N. SE. ENTER. '. One of my other problems, was the creation of the ever popular wandering monster bit, which may be of use to a new writer, especially a 128 owner, so these routines first tell whether the wandering whatnot is in the same location as the player, and then generates a random movement pattern (or any other action, such as combat) for it! So first of all in process 1 just after the..

```
* * 0 AT 0 ANYKEY GOTO 'start room', etc
```

We need an action which finds and holds the moving P.S.I's present location..

* * ZERO 31 COPYOF 13 65 .

If this is the very first time the enter key is pressed in a game, the next action makes a copy of the P.S.I's present location into a blank flag (flag 65) and this next action..

* * RANDOM 68

Randomizes another flag, from which the random directions are taken, and is then used as the number of the verb to move the P.S.I in (see: LET 33 ?/MOVE 65 for printing exits on screen). This must 'not' be included in the ZERO 31 routine. This next piece tells if object 13 (which I've taken as our wanderer) is in the players location, and then goes onto a process dealing with its movement, etc.

* _ PRESENT 13 PROCESS 5 DONE

With me so far? Good, then for process 5 we need to handle the finding of the direction the P.S.I is to move in, and then the printing of its direction on screen. So is the random number generated in process 1 less than 15 (ie, a direction?)

* _ 0 LT 68 15 PROCESS 6 MOVE 65 COPYOF 65 13 LET 51 13 SYSMESS F
PROCESS 7 DONE

If so, then the routine goes to the process dealing with the generation of the direction chosen by the random number, if the flag is able to move in the direction (sometimes it won't, if the random number isn't the same as a connection) given, and is thus the same as that of a connection, then the copy of the P.S.I's new room is copied back into the P.S.I's present location flag, which then moves the P.S.I in the direction randomly made, it prints sys F and then prints the direction the P.S.I moves off in. Sys F _ 'The _ wanders off '. Remember the space after 'off '. Now the reason I inserted LET 51 13 before the sys is just to allow this routine to be used for other P.S.I's if wished, as this is the flag number which prints the name of an object given in place of an underline in a message. Yes, but what if the random number is not available as a connection? Well that can be handled just like this..

* _ 1 LT 68 15 MESSAGE 1 DONE

The process goes to the routine above, if it cannot move the P.S.I.

* _ 2 GT 68 15 MESSAGE 2 DONE

* _ 3 GT 68 24 MESSAGE 3 DONE, etc

I've left message 1, 2 and 3 upto the writer, and as many exist as the writer wishes, just by measuring whether the random number is greater/less than a value specified, and now onto process 6 which generates the random number needed, and as before one exists for each available direction, except for the last one which is taken care of without a chance option, as it is the last one in the process (ie, nothing comes after it)

* _ 0 CHANCE 10 LET 33 2 DONE

* _ 1 CHANCE 10 LET 33 3 DONE, etc upto..

* _ 12 LET 33 14 DONE

These use the conversion nouns from 2 - 14 (or less if the writer wishes to give the player less directions)

This is the last process in our sequence of three (process 7) and is the process dealing with the printing of the direction the P.S.I has moved off in, and again there's one for each available direction.

* _ EQ 68 ? MESSAGE ? DONE

Is the random number used in the movement of the P.S.I the same as an available direction? Again this uses the conversion nouns from 2 - 14, if so then the relevant message is printed, for example 'N. ' for north, 'U. ' for up, etc. Here is also a nice little routine of mine (which Larry Horsfield may (or may not) have used in the 'Probe' (thanks Larry!)), to calculate whether one flag is greater/less than another (something the P.A.W is sadly lacking), and is best used in a process table called from response when needed..

* _ CLEAR A SAME B C NOTDONE

So first of all we need to clear the contents of a chosen flag (A) to use as a counter, and if the two flags you wish to compare are the same then the process moves onto this next routine.

_ _ LET A 1 SUB B C ZERO C NOTDONE

Now if we make counter A equal to 1, and then take the contents of B away from C, if C is zero, then B is obviously greater than C!

_ _ LET A 2 NOTDONE

But if C is not zero, then it comes here and makes A equal to 2, as B is then less than C, ie, if A equals 1 then B is greater than C, but if A is equal to 2, then C is greater than B, which could be used to test if one characters might is stronger than anothers like in a fight, for flags B and C could both be random, but I'll move onto some fight routines next month, so here's where to get in touch for any help you're in need of, or for any hints you can give to other people, also please enclose a S.A.E if you want a personal reply, otherwise it'll get printed here instead..

G. E. MARCH, 93 ROBERTS STREET, NEWCASTLE UPON TYNE, NE15 6BE



FOOD CHART

Devised by C.Hester



- 1) "Baby I Love Your Whey" - Will To Powder
- 2) "Don't you Forget About Meals" - Simple Minds
- 3) "Thank you For The Muesli" - Abba
- 4) "Manic Sourtle" - The Bagels
- 5) "I Want To bake It With you" - Bread
- 6) "Dark Side Of The Mousse" - Keith Floyd
- 7) "Souperly Guy" - Eggs-Express
- 8) "I Just Diet In your Arms Tonight" - The Cooking Crew
- 9) "Strawberry Flans Forever" - The Eaties
- 10) "Foot To Fry" - The Rolling Scones

Machine Coding your adventures Part 1

by Paul Brunyee

Perhaps some would call this article "Masochistic Adventure Writing" or "Blowing your mind in three easy steps" but what I am going to do is show how easy (!) and more importantly, how individual and original your adventure can be when written using assembly language.

I will start by showing how some of the building blocks of an adventure can be written. By building blocks I mean such things as command input and command parsing routines. Such routines may then be incorporated into your own BASIC or assembly language programs.

I must point out at this stage that I program using a Z80 based assembly language (on a Speccy actually) so my apologies extend to all you 6502ers as I am only familiar with Z80 coding.

But why assembly language? Well, quite an easy question for starters. The processing involved within an adventure, such as decoding and acting upon your commands, is quite large, and consequently rather time consuming when achieved through BASIC, or any other interpreted language. The problem, in this context, is that with interpreted languages the CPU has more work to do breaking down the high level language statements into code it can act upon when the program is executed. With an assembly language program, before execution it must be translated into machine code by a special program called an Assembler. When it is translated, or assembled, in this manner, the output from the Assembler (the machine code) can be directly executed by the CPU. Thus, an assembly language program, once assembled, can be subsequently executed without any further processing involved. This leads to a significant reduction in processing time which, in adventurespeak, means faster response times.

There are other advantages gained with using assembly languages, such as more compact and efficient code, although the speed factor is probably the most important.

An assembly language is a programming language whose instructions are written in a mnemonic form, or a form in which their sound suggests their meaning, as in the following examples:

```
LD A,27  - load accumulator with decimal 27
INC A    - increment contents of accumulator by 1
CP 72    - compare contents of accumulator with decimal 72
```

An important feature of assembly languages is that assembly language instructions are based on a one to one relationship with machine code instructions. For example, to take one of the above examples, INC A is represented in machine code in one byte as decimal 60, whereas in a BASIC program a statement such as LET A=A+1 would translate to several machine code instructions, and be several bytes long.

As I mentioned briefly, before an assembly language program can be executed it must be translated into machine code by a program called an Assembler. Machine code describes the binary patterns, or digits, which form instructions which the CPU can directly act upon. Most Assemblers include some kind of editor for you to code your program into. Once coded, the program is assembled and, voila, you have executable machine code.

For these articles to be of much use to you, you will need an Assembler yourself. I use Hisoft's DEVPAK. I will also assume you have a basic understanding of assembly languages, but for those who are quite lost when talk of bytes, floating point arithmetic and hexadecimal representation fills the air I will willingly help, as best I can, either through these pages or on a personal basis, according to demand.

Z80 Overview

Within Z80 assembly language, there are seven main registers or variables, each only one byte long and thus able to take values from 0 to 255. These registers are the A, B, C, D, E, H and L registers. The A register is known as the accumulator and is used for most arithmetic and logic operations. The other six registers may be paired together as BC, DE and HL and in this register pair form are two bytes long, and thus able to take values from 0 to 65535.

When writing in Z80 assembly language an important note to bear in mind is the use of ROM calls to ease the programming effort. After all, if a routine already exists then that can be used instead of writing your own. The main advantage is the memory saved by using a 3 byte CALL instruction, rather than duplicating any code. I will be using several ROM calls in the coding.

I could write at length about the syntax and use of the Z80 instruction set but this may detract somewhat from the objective, actually writing some code so I will now list some commonly used pieces of code.

Open channel 2 - this is necessary to print on lines 0 to 21:

```
LD A,2      - set accumulator for channel 2
CALL 5632    - call 'open channel' routine
```

Print a character:

```
LD A,x      - load accumulator with ASCII character code
RST 16      - call 'print character' routine
```

Print a string of characters (long method):

```
LD B,9      - set for 9 characters
LD HL,STRING - point HL at start of characters
LOOP LD A,(HL) - pick up character at (HL)
RST 16      - print character
INC HL      - point HL at next character
DJNZ LOOP   - decrement B register, if non-zero jump to LOOP
RET         - return to system
```

STRING DEFM "Adventure"

Print a string of characters (short method):

```
LD DE,STRING - point DE at start of characters
LD BC,9      - load BC with number of characters to print
CALL 6252    - call 'print characters' routine
RET         - return to system
```

STRING DEFM "Adventure"

These last two pieces of code help show the usefulness of calling upon routines already in ROM. Both routines achieve the same results but the second gives a saving in memory requirements.

The calls so far shown are only applicable for the Spectrum but the principle is exactly the same for other Z80 based machines, such as the Amstrad. The only real difference from machine to machine is the address specified as part of the CALL statement.

Note that use is made of a DEFM statement which is not an assembly language instruction but an assembler directive and is used by the assembler program when assembling the source code. The DEFM directive tells the assembler to reserve space in memory equal to the size of the operand, in our case 9 bytes, and define the contents of this space to be set to the ASCII representation of the operand, in our case "Adventure". Thus, 9 contiguous bytes following the RET

instruction would be set to the values 65, 100, 118... and so on. These are the ASCII equivalents of the characters "Adv..." etc.

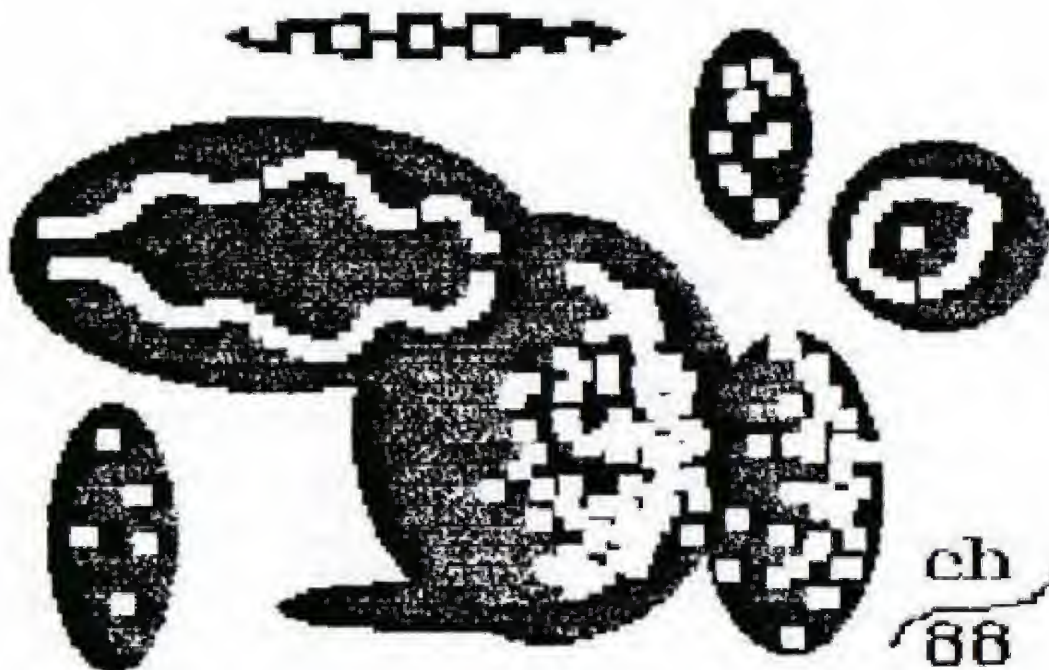
Other commonly used directives are ones such as DEFB, for 'define byte', and EQU, for 'equate label with operand'. As I use each of the directives I will include an appropriate comment to describe it's function.

Reading the keyboard is, of course, a prime concern and on the Spectrum a technique for this is to continually check bit 5 of the FLAGS system variable (at address 23611). Bit 5 of this address is set by a ROM routine which is called automatically. When a key is pressed, the ASCII code for the key press is stored in the LAST KEY system variable (at address 23560). The following code can thus be used to read the keyboard:

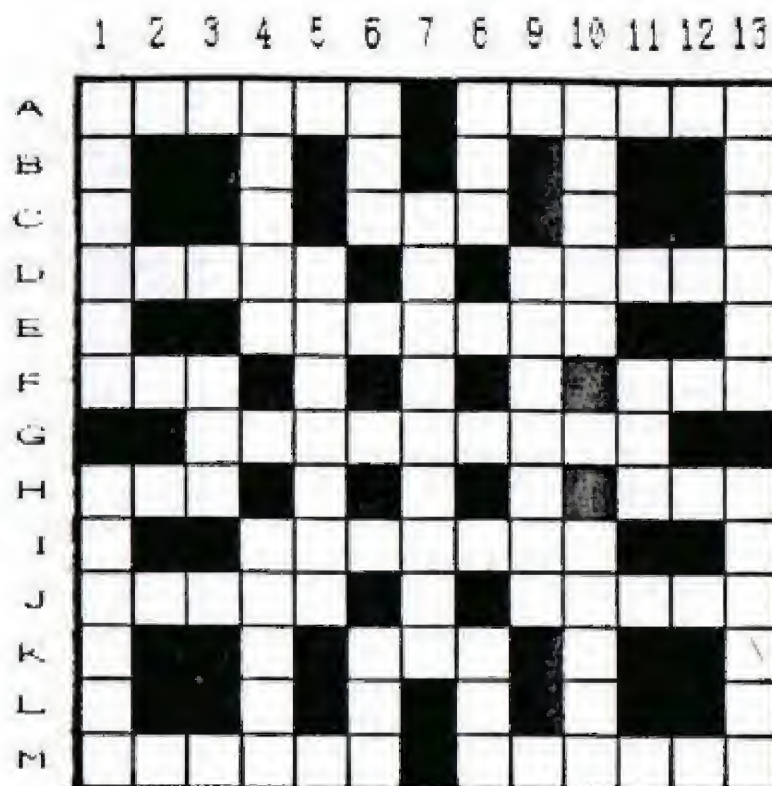
```
RES 5,(IY+1) - reset bit 5 of FLAGS
               (IY is a register pair normally holding 23610)
WAIT HALT      - force Spectrum's KEYSOAN routine to operate
BIT 5,(IY+1) - test bit 5 of FLAGS
JR Z,WAIT      - jump back if flag not set
LD A,(23560) - store ASCII value of key press in accumulator
```

Next moon I will start the design and coding of a command input routine and by that time I should be able to offer some Amstrad specific routines as well.

I welcome any queries and comments you have and will help with any problems as best I can.



ch
88



Crossword devised by C.Hester

Crossword

HORIZONTAL CLUES

- 21** Searches into copies of a popular adventure magazine (6)
38 written with incentive (6)
46 It's a plus with Maths (3)
51 Absence of the singular (2,3)
59 Existing creature (5)
64 Lowest pointed mark (3,4)
71 Finish together a knot (3)
711 LARD before athletics (3)
83 Agatha Christie play for doomed peripheral? (9)
88 Very small amount of money - one p short to buy a hot bowl? (3)
111 Easy talkative American petrol (3)
14 no exits! (7)
11 Check this for your progress (5)
19 Kingdom (5)
26 Jar for ashes (3)
11 Average sportsport in the sentinel? (6)
18 School exam for Ingrid's creators minus 9? (1,5)

VERTICAL CLUES

- 11** Forbidden Intocom title in Autumn? (6)
44 Home of the Three Bears and more (5)
46 Neptune's delight (3)
58 Deity (3)
110 Saddest and loudest person in the town (5)
113 Fabled beastie turned computer (6)
27 Angry Bomb The Bass hit without the Beat? (9 or 3,6)
15 Ask further information (7)
29 Obstacle (7)
23 Rod Hull's infamous bird (3)
111 L+D genre taking over from adventures? (1,1,1)
11 Method (6)
113 Crashing icon (6)
14 leach skills on the rails? (5)
110 Thick (5)
26 All objects are red herring? without a --- (3)
28 Novel prefix (3)

Whatever happened to..?

Each month I'll be taking a look at programs the companies left on the shelf. Adventures that failed to appear; utilities that were never completed. If you've any suggestions for this part of the magazine, why not write in and let me know? Maybe there's a program you've been anticipating for a long time that seems to have been forgotten about.

Valley Of The Source

To start with, remember a classic adventure that had many BBC owners raving over the graphics? Yes, that's right. A BBC adventure with graphics!! A seemingly rare beast and for once BBC owners weren't forced to watch their Commodore and Spectrum owning friends enjoying games with graphics that, when converted to the memory-tight Beeb, always came out text-only. The game? "Twin Kingdom Valley" from Bug Byte! The graphics employed a unique compacting routine that occupied about 8K yet was able to give hundreds of large and colourful pictures to all the locations. The trick lay in using a set of base graphics, such as a chair and a table, and repeating these throughout the game, but in differing sizes and combinations. If you've seen "The Illustrator", it's very much like the way you can move and scale an object, all from one initial drawing. Well, wasn't "TKV" a good game? What better than to release a follow-up...

Since "TKV" was released on the BBC, it became subsequently available on the Spectrum and Commodore 64 too. Owners of the latter machine looked set to be in for a treat (and probably other computers too) when in early 1985, plans were revealed for a follow-up to "TKV" to be called "Valley Of The Source". The man responsible was Trevor Hall who was claiming the game would feature half a million locations! Wow!!! The player would have to search through these (what, all of them?) in order to find the source of the River Of Gold from "TKV". Amazingly, each location would have four views in full-screen size, like "Lords Of Midnight". There would also be animated sword-fights possible between the many interactive characters. All sounds a bit like "Derender Of The Crown", but don't forget this was 1985! The game was set for release at the end of that year, but only on the Commodore 128 due to the many astounding features programmed in.

It is now four years since the game should have been released. I feel cheated when such incredible claims are leaked about a forthcoming game which then fails to appear. Where is it? I want my "Valley Of The Source"!! Imagine all those 500,000 locations to explore - months and months of adventuring to enjoy! Each location graphically depicted! Surely a winning program with so many stunning features? But that's just it. There must have been too many features for the good of the game. Imagine if you were writing an adventure on the Commodore 128. You'd barely have enough room for all the locations, which would have to be compressed or course. Let alone pictures of them, never mind a game as well! Now take a hundred locations and you could do a game in 32K! Half a million would take more like a super-computer! But memory's no problem if you've access to a disk drive, you simply store as many parts of your game as you like onto as many disks as you like! Infocom games aren't limited by the internal memory of the computers they run on, the text is stored on disk, waiting to be called up whenever necessary. Hence the abundance of text in an Infocom adventure. Now had it been Infocom instead of Bug Byte attempting "Valley Of The Source" and I reckon we'd have seen the game in the shops as scheduled in 1985! I also have no doubt that a lot of people would have bought it on the strength of "Twin Kingdom Valley".

Next month...

Hopefully, here's what's in store for Issue 2 of Adventure Coder - out mid-August!

- * the first letters page - that is, unless no-one writes in!
- * an article on advanced PAW routines from a top adventure company
- * less of me, more of what you've written, so get writing and see your name in print...
- * GAC and GAC+ pokes for the Commodore 64 to alter the screen colours
- * a deeper look at GAC generally and the bugs to avoid
- * some SLAC routines for SI users
- * another superb story
- * more amazing artwork (?)
- * a look at another program that never got released - will it be a utility or a game?
- * a further devious crossword to baffle you, plus the answers to this month's
- * the second part of Paul Brunzee's fascinating article on machine coding your adventures
- * more PAW routines in George March's PAW Prints column
- * another wacky zany chart - what'll the subject be next month?

All this, plus a host of other items in the next issue of Adventure Coder, only £1!



GAC to the drawing board

by Chris Hester

Graphics in adventures? Who needs 'em? But you've bought Incentive's "GRAPHIC Adventure Creator", so you'd better make use of the facilities it offers. Besides, without pixel-pretty pictures, your mega adventure won't be worth a thing in the big bad professional market out there, where no piccies means no reddie's, like it or not. Level 9 have 'em, Magnetic Scrolls have 'em, and now even Infocom too! So, unless you're satisfied with selling a text-only game on the home market (and why not?) then graphics are a must. GAC doesn't score too badly here, though it's far from brilliant. I hope to show you just how to get the most out of what's on offer.

Colour

The right colours are essential to a successful graphic. Try and choose colours that fit the subject you're drawing. A garden would probably need GREEN, for the grass, and a BLUE for the sky. If the text for your garden describes a prominent tree, then that should also be visible in your graphic, so choose a wood colour such as BROWN or ORANGE for the tree. But wait! Trees can also be GREY or WHITE and many other natural shades, so you're not limited to just brown. Don't choose a bad colour, though - blue would be most unsuitable for a tree! Since you've only four colours to use on the Commodore 64 and Amstrad versions of GAC, try and choose four colours that not only give you enough to draw your whole picture, but also look good and mix well. Blue may be fine for a sky and orange fine for fruit, but come to shading the background and you may find a shade from these colours looks revolting, so you'll have to alter your blue to cyan to get a decent shade. With the Commodore 64, blue and orange give a truly disgusting shade, rippled with undesirable lines. Experiment by cycling through the colour bar at the side of the screen until you get a good mix. Shades can make or break a graphic, so get wise to decent combinations of colours as you can use them again and again in other graphics, even selecting colours for a good shade to start your graphic with. Shades are essential to getting beyond the restrictions of just four colours. (They can be a two-colour dotted mix using any of the four colours.) For instance if your graphic uses red, yellow, green and blue, straight away you might realise that red and yellow will give a shade resembling orange. Also, yellow and blue will give a green, so there you might not need to select green itself as one of the four colours! That gives you the chance to select another colour, say purple.

There aren't any set rules for colouring objects, you have to learn from experience. Often what you think will be a suitable colour doesn't somehow look right on screen and you may find an unusual colour works better. As an example, you may have drawn a room with the doors visible and decided to have these white. However, this may look too harsh and cold - perhaps your room has a fireplace inside it? If so, selecting yellow instead of white may give the picture just the right warmth of colour and not look unusual with yellow doors. At least you're not painting in oils, where you'd be stuck with a colour once chosen unless you painstakingly went over it with another colour! On the computer you're free to alter your colours as you wish, until you get them right.

Another point of advice is to avoid overuse of garish and gaudy colours. You may think they look good as colours on their own but the resulting graphic might well look a horrendous mess. Colours around us are generally a lot duller than those on a computer screen - a walk in

the park would present you not with bright red, bright green and bright orange, but a whole range of greens, some quite dark, and various shades of browns that are beyond cheaper computers. Even black isn't as jet black as you get with a computer - due to light from the sun, or a lamp in your room, a black ball will appear grey ontop and dark grey on the sides. Only near the bottom will it appear dark black. There might also be a white highlight, a reflection of the lightsource such as an electric bulb visible. Black? Not quite. The Commodore is blessed with three shades of grey as well as black and white in it's colour palette, so objects can be made to look very realistic using shades of all three greys. You may be writing a historical adventure, perhaps set in the Twenties - what better than to use "black and white" pictures in your game using white, light grey, dark grey and black. It would look very atmospheric!

Colours are in the end personal and no one can say such-and-such a colour will be right for your game. You must experiment and learn which colours work best and which you like yourself. Most of all, have fun!

Perspective

This is an extremely useful tool for designing your graphics. Basically, perspective is a way of showing objects in the most realistic way possible, so that objects far away appear smaller than those nearer you, and at the correct angle. Perspective works on a grid which stems from a focal point. This is the first thing to establish. The focal point relates to where you are looking at a scene. This determines how your objects will appear in three dimensions. Perspective is a very realistic approach to use and you may find it too real to use, preferring a more artistic method of drawing your graphics, but I doubt if they'll be exact and I've seen many a game where the laws of perspective have been broken so badly, objects in the same room appear to be facing completely different ways, as if stuck on the screen and not in the room depicted at all! You can't go wrong with perspective because once your grid is set up, everything simply fits onto that. To set up the whole grid, decide where your focal point will be then. Are you looking at the view from a doorway into a room? If so, place the focal point half-way down the screen in the middle. You might like to place a dot as your first graphic command since the cursor starts bang in the middle of the picture, but maybe you're looking down from a castle over a village? Then the focal point must be placed high up, but not necessarily in the middle of the top line. You might be looking out of the left turret of your castle and it might be an idea therefore to set the focal point off to one side a little, as if you were viewing the scene from the side. If you're looking instead at the castle gates from the wetness of the moat (having fallen in!) then place the focal point on the bottom. Having decided upon the viewpoint for your picture, place a dot there. From that dot, most of your lines must stem!

I find that a lot of objects start off best as a simple box shape from which you can go on to alter that shape to the object required and add detail. Before you put in any objects though, you must think about the space they will occupy. Are they in an open field or a small room? Can you see the whole of the room, or are you at the side of it? When I wrote my last adventure, "Runaway", for my Commodore 64, I set it inside a house which meant that each location was inside a room. For convenience I also made most rooms the same size. The player saw the whole room in the graphic, which meant a left wall, a right wall, a back wall, a ceiling and the floor. This was like looking into a rectangular box. The focal point was consistently in the middle of the screen as well. From there, I drew four lines out from the middle dot

to the corners of the border. This gave me the walls, ceiling and floor though it looked like the room went on for infinity, so a back wall was needed. This was set up by placing a large rectangle onto the lines. This is where a grid comes in immensely handy. Drawing the room freehand would mean you'd have to use your judgement to place the back wall accurately. However, with a grid of lines drawn to the middle focal point put in first, the back wall has to fit onto the four lines, or it'll simply be wrong. Once the back wall was put in, the room was complete. This box-shaped room could then be called using the Picture command at the start of every new location. But what about objects inside the room? My first object to insert was a bed stretching from the left wall to the middle of the room. Here again, perspective came in immensely handy. To draw the headboard of the bed, all I had to do was draw a line out from my focal point half-way up the left wall, followed by a straight line down to the floor. Again, the bed could be visualised primarily as a box-shaped object. The trouble comes when you find that you've drawn the lines to make up a box but it looks like one of those wire-frame models. The trick is to eliminate the lines that won't actually be visible, i.e. the back of the bed and also the edge of the floor that the bed covers up. To do this requires you to go back over your lines only drawing those that'll be seen from the viewpoint of the picture, i.e. the front of the bed, the remainder of the wall and floor, etc. If you find this a little complicated, then select a different ink colour to show which lines you've gone over - you can always alter the ink colour again afterwards. I found it best to step back to the original lines after going over them and deleting each line as I worked round the object. This meant that when you got your whole picture to draw up afterwards, only the correct lines came up. The back of an object would not be drawn up, as you'd have deleted it. Note that many lines are simply deleted away from the wire-frame object and not gone over again. You're not duplicating your work going over the lines, you're actually selecting only the parts of the wire-frame that you choose to be seen. You may find that some lines end up with a fraction of their original length, yet that bit may be necessary to complete part of the object. When all this is done, you'll find the object looks incredibly realistic and accurate in it's setting, as you can see from the bed and the wardrobe in figure 1. The wardrobe was set up again as another box and it's hidden lines removed, that the viewer wouldn't see. Note that part of the original box shape of the room also gets deleted, and you can see how the bottom corners of the back wall have become hidden from view. This requires you to step back afterwards and delete the back wall rectangle, having gone over the remaining visible lines around it. It may sound a little complicated at first, but you'll soon get the hang of things and be creating your own graphics that look just right, call them 3D graphics if you like!

Ellipses

These are the two most useful commands in the graphics section of GAC and you'd be surprised how versatile they can be. Ellipses can be stretched horizontally and vertically giving rise to circles for a start. What better shape to use for a graphic? A circle can be made to be a sun in the sky, a wheel on a car, a crystal ball... if you're using a perspective grid as above then you'll find you can insert circles in your picture as if they're lying on the walls, simply by using an ellipse. Take a look at the mat on the floor in figure 1. To the viewer, this is clearly a round mat, yet it is drawn using two ellipses - in other words the circular mat has been stretched to lie on the floor. Can you spot any other ellipses used in the room? No? How about the wardrobe door handles! Amazingly they are drawn using just two ellipses, small enough to break up into pixels that can't show the curves any finer so the ellipses appear jagged. However, far

from this being undesirable it adds to the effect and creates what would otherwise take several lines and dots. Ellipses can be used to look like lines, thus saving vital memory in the computer. The handles on the two room doors look like they were drawn with a dot and a line. They are in fact also ellipses, that are so small they appear as crosses that overlap onto the edges of the door. Good, eh? See for yourself: the many shapes ellipses can create - the smaller they are the less like ellipses they become. There's even a way to chop off an ellipse when it looks like a circle to give a semicircle or less! This is shown in figure 1 at the corners of the border. There, an ellipse was drawn at the corners so that three-quarters of the circle would have had to be drawn outside the edge of the picture. Since this isn't possible, GAC flattens the circle along the edges, yet leaving the part inside the picture nice and round. Hence you get a quarter of a circle! This is useful for corners and shapes such as a setting sun - place the sun at the bottom of your picture and it'll flatten out to a semicircle.

Rectangles

One of the tricks I found here is to replace two adjacent lines with a rectangle. The edge of the wardrobe in figure 1 might be drawn in several ways - with lines and then filling the space inbetween, or with lines and adding a couple of inner lines to draw over the space. Both techniques require many lines, at least three. However, with just two rectangles you can create the same effect. Slim the rectangle down until the sides touch and you lose the middle space - you now have what looks like two lines together, which are really a closed rectangle! Place this down and it takes up the room previously needed by two lines. Open rectangles can also come in handy in replacing wasteful lines as in the door on the back wall. If it weren't for the bed, the door might have been drawn with six lines - three along the outer edges and three for the inside frame. However, instead of six lines, why not use two rectangles! It doesn't matter that a rectangle might overlay another line in your picture as you'll be saving bytes using rectangles instead of lines. One for the outer edge of the door, and one for the inner frame! Alas, once I put the bed in, I had to resort back to lines as the corner of the bed went over the door!

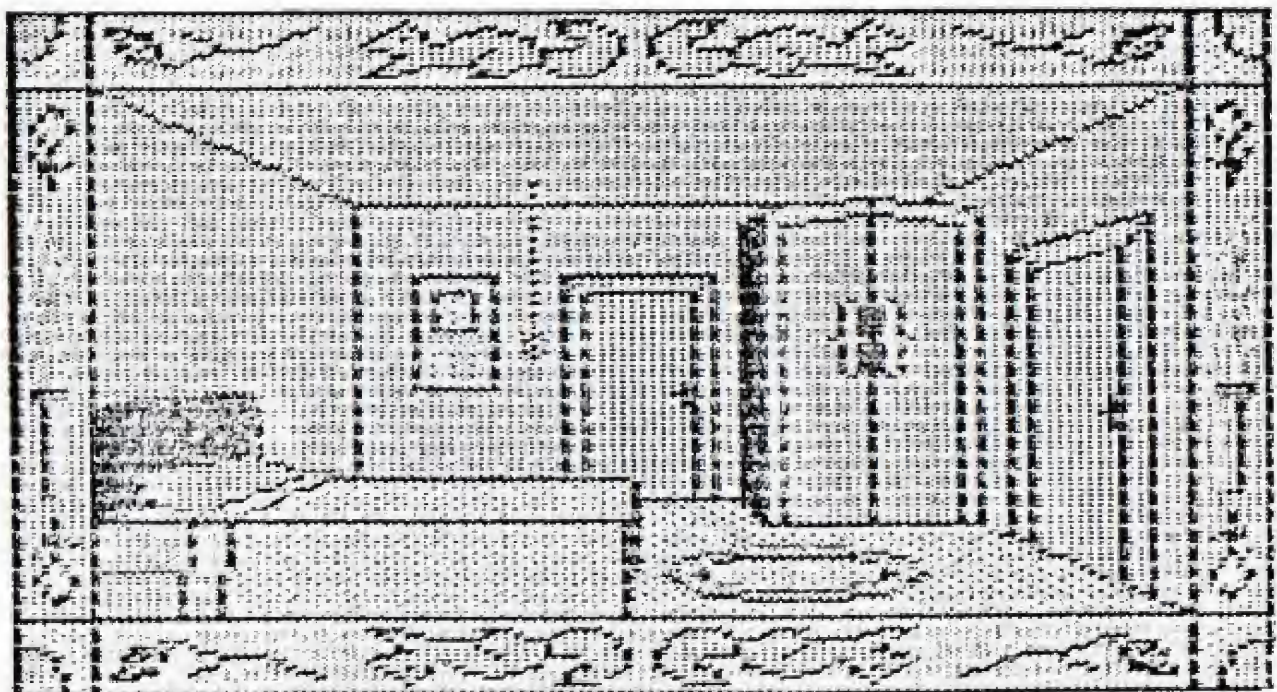


Figure 1: A graphic from "Runaway" showing use of perspective.

The Burning Man

(1989 Rewrite) - an original story by Christopher Hester (C) 1989

It all began one night in the warmth of a cosy living room. The year doesn't matter, nor the day. Judy lay comfortably on the leaf-green sofa, enjoying the heat of an open fire. She felt relaxed and easy after a busy day at the office. Her auburn hair fell down around her slim neck touching the cool pink dress she wore, with a lipstick red belt. Judy was in her early twenties yet she still maintained the youthful look she'd perfected at 19. She had never married and only dated occasionally. The relationships never seemed to last long. *What did it matter*, she thought, and stretched out, dropping off her white high heels into the carpet floor.

Flickering firelight softened the hues of the room, painting the walls with dancing shadows. *Just right*, she thought, smiling to herself. The fireplace roared and crackled, a small chopped log burning nicely on top of the glowing coals. Judy focused on the log, then the coals. *How hot they burn, as hot as hell*. The flames were an ever changing pattern, a play of fire and light. *I can almost make out glowing shapes...* thought Judy, studying the fiery flames from the coals. *Different shapes, like animals, birds, a face...*

A face. The face of a man. His eyes burning out at her, almost real. *Too real*, she thought. But the face didn't change. Judy blinked. Still a face. With a nose, and a mouth, lips on fire. The face grew clearer, the heat more intense? Judy watched flabbergasted as the face appeared to rise up from the coals into the lifesized head of a man.

"Do not fear me, young lady," spoke the fiery head, with a rough tone. "There is nothing to be scared of. You are a pretty sight." The head rose up, revealing shoulders that soon gave way to arms and a chest, until the blazing figure of a whole man appeared at the fireplace. His body burnt with a ferocious heat, great flames licking his entire body from head to foot. Judy lay transfixed upon the sofa, not daring to move, yet she felt strangely untroubled by the figure before her. He held out a hand, its fingers lost in the engulfing rage of heat sizzling the air.

"Come with me, young girl," asked the burning man. "You will know what I know..."

"And... and what is that?" she asked the man on fire, caution in her voice. Was she now dreaming?

"My life is the life of fire, the life of eternal heat. See this hand, this body? It is on fire, yet I do not burn away. I am proof that I exist, forever to live enflamed."

"Where... where do you come from?" she asked him fighting a desire to leap from the sofa and run, yet there seemed to be no reason to be afraid.

"Ah... that is a long story. I was once a human being, existing like yourself, with the inability to withstand strong heat. For I would suffer great pain from burns. But one day I was at home, sat in comfort, when I thought I could hear a noise outside the room like that of a bonfire, and I realised the house was indeed on fire! I struggled to reach the door and escape, but I let in the flames and the heat forced me back. I fell over, and tried to reach the window, but the fire had leapt to the curtains and the whole room was ablaze. I knew I was trapped. Hot flames licked at my flesh as smoke clouded

into my lungs. I collapsed and was burnt to death. But now I stand here, still burning. When you lit the open fire behind me, I was given a chance to reform my body here, in the strength of the flames. Fire was once my enemy, now it is my friend. For, you see, when I died, my soul was not allowed into heaven, but sent straight to hell. You want to know what hell is really like?" Judy remained silent, feeling a mixture of suppressed sympathy and fear.

"I'll tell you what hell is really like," he said burning to tell her. "Hell is... death. A constant death. However you die on Earth, you are kept dying in hell, never allowed to complete your death, your soul never to pass on to heaven. I died in fire, so hell for me is this fiery shape that you see before you. I am forced to burn like this forever... But hell is like any other place, it has a way in and a way out. I searched for many years to find a way out, in the hope that I could complete my death elsewhere. Eventually I came upon a tiny hole - an opening in hell that led down a deep corridor to where I knew not till now. That corridor led me here, to your fireplace, and yet still I do not finish dying. It seems I will always be the burning man." He stopped and peered down at his legs, the melting flesh that never melted away, engulfed in flames. A moment of sadness fell over the room.

"Perhaps it is not good that I have come here," he pondered outloud, "I'm afraid my presence has a rather nasty consequence for you." Judy sat up on the sofa.

"What do you mean?" she demanded.

"It is possible for me to remain on fire like this, but you are still a living being. Flames do not burn me, but they burn the things around me. Have you not noticed the carpet is on fire near my feet? Within seconds the whole floor will be alight, and your sofa surrounded by flames. There is no hope for your survival now. I'm sorry, it is too late." He looked around the room as the flames spread across the floor and up the curtains. Judy screamed and leapt up in the sofa as fast as she could, but her dress was alight, flames and smoke driving up her body. Within a minute she was burnt to death. When the flames had engulfed the entire room, the burning man returned to the fireplace and stepped back inside the corridor to hell. He did not wish to return, but he could do no good on Earth. Yet he thought it might not be as bad in hell as it had been before... At the end of the corridor he saw her, on fire just like himself. Judy took his hand and they walked through hell together.



Useful addresses

This list is intended to help you in selling a game and getting it reviewed. Often you may know the name of a company that's published adventure games before, but you can't find the address anywhere! Well help is at hand here. What's more, where possible I've also included the telephone numbers of each company incase you want to ring and ask them what's happened to the game you sent in. Even if you're merely after additional information on a company's products, stop searching madly around for an address, it'll probably be here ready for you. If you have any other addresses you've found useful in the past, let me know and I'll include them in future issues.

AMI = Amiga 500, (maybe 1000, 2000 too)
ARC = Archimedes (30E, 310, 310M, maybe A3000 too.)
BBC = Acorn BBC Micro
C64 = Commodore 64 (maybe 64X, 128, 1280 too)
CFC = Amstrad CFC (464, 664, 6128)
ELE = Electron
S48 = Spectrum 48k (maybe 16k, 48k+, 128k, +2, +3 too)
ST = Atari 520STFM (maybe 520STM, 1040STF, MEGA 1, MEGA 2, MEGA 4, STE too)
VAR = various computers

ADVENTURE MAGAZINES

VAR: Adventure Coder:- Christopher Hester, 3 West Lane, Baildon, Nr. Shipley, West Yorks, BD17 5HD.

VAR: Mandy Rodrigues, Adventure Probe, 24 Maes Y Cwm, Llandudno, Gwynedd, LL30 1JE.

ADVENTURE COLUMNISTS

VAR: Steve Cooke, Ace, Priory Court, 30-32 Farringdon Lane, London, EC1 3AU.

AMI: Dave Eriksson, Amiga Computing, Database Publications Ltd, Europa House, Adlington Park, Adlington, Macclesfield, SK10 4NF.

CPC: The Pilgrim, Amstrad Action, Future Publishing Ltd, 4 Queen Street, Bath, BA1 1EJ.

ST: Brilling, Atari ST User, Database Publications Ltd, Europa House, Adlington Park, Adlington, Macclesfield, SK10 4NF.

C64: Andy Moss, Commodore Computing International, Croftward Ltd, Finsbury Business Centre, 40 Bowling Green Lane, London, EC1K 0NE.

C64: Gordon Hamlett, Commodore Disk User, Argus Specialist Publications Ltd, Argus House, Boundary Way, Hemel Hempstead, HP2 7ST.

C64/AMI: Keith Campbell, Commodore User, Priory Court, 30-32 Farringdon Lane, London, EC1 3AU.

VAR: Keith Campbell, Computer + Video Games, Priory Court, 30-32 Farringdon Lane, London, EC1 3AU.

ELE: Fendragon, Electron User, Database Publications Ltd, Europa House, Adlington Park, Adlington, Macclesfield, SK10 4NF.

VAR: Paul Rigby, The Games Machine, PO Box 10, Ludlow, Shropshire, SY8 1DB.

BBC: The Mad Hatter, The Micro User, Database Publications Ltd, Europa House, Adlington Park, Adlington, Macclesfield, SK10 4HP.

S46: The Sorceress, Sinclair User, Priory Court, 30-32 Farringdon Lane, London, EC1 3AU.

S48: Mike Gerrard, Your Sinclair, 14 Rathbone Place, London, W1P 1DE.

C64/Ami: Prof Norman Nutz, 22AP!, PO Box 10, Ludlow, Shropshire, Sh10 1Db.

BUDGET ADVENTURE COMPANIES

VAR: Alternative Software Ltd, Units 3-6, Baileygate Industrial Estate, Pontefract, West Yorkshire, WF8 2LN. Telex: 557994 RR DISI G Fax: (0977) 790243 Tel: (0977) 797777

VAR: Mastertronic, 2-4 Vernon Yard, Portobello Road, London, W11 2DZ.

VAR: Rack-It, Hewson Consultants Ltd, 56B Milton Park, Abingdon, Oxon, OX14 4EX. Tel: (0235) 832939

S48: John Wilson, Zenobi Software, 26 Spotland Tops, Cutgate, Rochdale, Lancashire, OL12 7NX.

ADVENTURE UTILITIES AND/OR ADD-ONS

S48: Camel Micros, Wellpark, Willeys Avenue, Exeter, Devon, EX2 8BE.

VAR: Incentive Software Ltd, Zephyr One, Calleva Park, Aldermaston, Berkshire, RG7 4QW. Tel: (07356) 77288 Fax: (07356) 6940

VAR: Gilsort International Ltd, 2 Park Crescent, Barry, South Glamorgan, CF6 8HD. Tel: (0446) 732765

S48: Gerald Kellett, Kelsoft, 28 Queen Street, Stamford, Lincolnshire, PE9 1QS.

S48: Simon Kimberley, Simicro, 115 Yelverton Road, Coventry, CV6 4AG.

CASSETTE DUPLICATORS

J&S Records, Freepost, 19 Sadlers Way, Hertford, SG14 2BX.

McGregor Tape Services, 42 Anchor Avenue, Paisley, PA1 1LD.

Simon Stable Productions, 20 West End, Launton, Oxon, OX6 0DF.

STATIONERY, PACKAGING AND PRINTING

Launton Press Ltd, Wedgewood Road, Bicester, Oxon.

Millway, Chapel Hill, Stanstead, Essex.

S&H (Processing) Ltd, Gotts Road, Wellington Bridge, Leeds, LE12 1ES.

SO WAT'Z
HAPNIN? DONT
ASK, I DONT
NO YET. SO
U'D BEST HANG
AROUND + C!

LETZ C...

JUST
A
ROLLIN'
BALL...

CUD THIS
B SUM
THING
NU?
OR
BOROED
+ BLU?

HERE
WE
R

BUT WARE
IZ THAT?
I MEAN,
WAT NEXT?

©1989 CHRIS
HESTER

LIKE...
A BALL!

SO WATZ UR
STORY, BALL?

HOW CUM
U UP A
TREE?

I'M
JUST A
ROLLIN'
BALL...

WELL
HEY!
ROLL
WITH
ME!

AND SO THEY
ROLLED 2GETHA
UNTIL THE
END OF...

...THIS STRIP!

Adventure Coder needs you!



Without your support, there is little to make up a magazine. I'm looking for contributions on any aspect of adventure writing, in particular these things:-

- * 16-bit utility routines (such as those for SIAC etc.)
- * anything on converting adventures between machines
- * any artwork you might have drawn - how about drawing the next cover?
- * anything on how you yourself set about creating an adventure - we all have our own ways - do you start with a definite plan, or let it grow from an idea?
- * short-stories to form a break from the other pages.
- * cartoons!

Never feel what you have to offer isn't good enough. The only reason I can see for material being refused is if something very similar has also been sent in that month, or there's a lack of space - in which case I'll fit it in the following month! However, to make it easier for me to use what you've sent, there are a few points to bear in mind, though you don't have to follow these as golden rules.

1) Try and use A4 sheets for your articles, drawings etc. This is because Adventure Coder is pasted up on A4 sheets and then reduced to A5 format as you have it in your hands now. It doesn't really matter if you have something on another size of paper, but A4 is the ideal size to use.

2) When posting your material, the best way to send it is by softly folding the A4 sheets once down the middle. They then fit inside an A5 envelope. Any further folds and you get too many creases in the pages which have to be flattened out when stuck down for printing. Please refrain from folding your pages every which way and cramming them in a tiny envelope!

3) If you own a typewriter or wordprocessor, it saves me a lot of time if your writing is already typed up ready for inclusion. Try and use a decent margin like you see on these pages - about a centimetre all round will do fine. Don't worry if you've written something in pen though - I can soon type it up! (For wordprocessor users however, I have my program set to a Left Margin of 5 spaces plus 2 at the top and 5 at the bottom. The number of characters across the line is set to 70 and there are 70 lines down the page too. Note though that your computer system may print text larger or smaller than mine so these figures may be of no use. Then again you might find them invaluable in sending in identical copy like you see here. Also I always print using Near Letter Quality (N.L.Q.) but don't worry if your printer doesn't have that, just make sure your print isn't too faint. I've set the computer to print a page that fits inside a 10cm border with a 22cm bottom area where I later add the page numbers. You could try printing different widths to get a similar result to me.)

SUBSCRIPTIONS

Price per single issue:

£1.00 United Kingdom
£1.75 Europe (+ S. Ireland)
£1.75 Rest of the world (surface)
£2.25 Rest of the world (air mail)

Price per 12 issues (1 year):

£12.00 United Kingdom
£21.00 Europe (+ S. Ireland)
£21.00 Rest of the world (surface)
£27.00 Rest of the world (air mail)

All payments must be made in pounds sterling. Cheques should be crossed and made payable to "C HESTER" please. (This also applies to the costs below.)

ADVERTISING RATES

£5.00 Inside full page (black-and-white A5)
£9.00 Inside double-page spread (middle pages)
£6.00 Outside back cover (on coloured paper).
£2.50 Inside half-page
£1.25 Inside quarter-page

Line ads 3p per word.

LETTERS

Letters sent to the magazine may be chosen for publication unless marked "Not for Publication". If you require a reply other than printed in the magazine, please enclose a stamped self-addressed envelope.

COPYRIGHT

Copyright on anything which does not carry the writer's name belongs to Adventure Coder. Copyright on all other items lies with the writer as Adventure Coder does not pay for contributions. Readers whose work is printed in the magazine continue to hold copyright on all material written by them and are free to use it elsewhere.

ABOUT ADVENTURE CODER

The opinions expressed within this magazine by individual writers are not necessarily shared by the Editor. Adventure Coder does not recommend or condone any particular firm or product.

Issues of Adventure Coder come out during the middle of each month. The deadline for contributions is the 20th of the previous month.

Software and hardware is greatly appreciated for review purposes no matter what computer it is for. Adventure writing utilities are especially welcome.

Adventure Coder is produced on a Citizen 120B with a Commodore 64 computer using a variety of programs.

EDITOR: Christopher Hester, 3 West Lane, Baildon, Near Shipley, West Yorkshire, BD17 5HD, England.

